

SAM farm upgrade

home page: <http://fnpcc.fnal.gov/SamFarm/SamFarm.html>

- SAM metafile platform
- CAF platform
- H-stream prototype (Elliot)

- Dispatcher submitting Exe jobs
 - Failure control and retry
 - Planned project with predefined jobs
 - Cut the threads, throw away the book, how?
- Dispatcher submitting Concat jobs
 - Planned project with predefined jobs
 - Concatenation with parenting records
 - Failure control and retry

- The book : SAM metafile
- Web link to project task control

Goal for SAM farm

- Migrate to SAM metafile platform
- Make CAFs farmers
- It is a potential GRID
- Make auto resubmit >10k/day submitter
- Tolerance/ less human intervention on book keeping
- Worry about IO later?!

What farm has CAF doesn't

- Dispatcher accessing run records
 - Dedicated IO to Enstore
 - Concatenation
 - Task control
 - Writing to the book
-
- farm is a small fraction of CDF computing
 - farm is the only platform read/write to DFC

What we learned with the farm

- Threaded file-tracking for concatenation
- Chained processes
- Assume no failed job

- Result: one cart derails, all fall
- Human intervention on file-tracking
- It is beyond human capacity >10k records/day

A better book

- Can not mess up the book !! How?
- Dispatcher has planned tasks (how?)
- Jobs has parenting records
- Failed jobs (how&why) get resubmitted
- SAM book is the book and only
- Task ends till child is on the book

CDF production farm capacity

- CPU: 300 pentium, newest 64 are dual pentium4 2.7GHz
12 M events/day peak observed
- dfarm capacity
23 TB, including 3 file servers (2Tb)
2 TB/day peak writing (x2 replica)
- I/O to enstore, Glinks
8 inputs, 8 outputs/concatenators (expendable)
one data-set writing to Enstore :
4M events (1k files, 700GB)/day peak

Enstore

Tape Library

Tape drives

Movers

100Mbit/s

Fcdflnx3

CDF Switch

1Gbit/s

Cdffarm1

1Gbit/s

Cdffarm2

Farm Switch

Fnpcc

1Gbit/s

1Gbit/s

100Mbit/s

Cdfora1

Fcdfsgi2

Readers Writers

Worker Nodes



